An Energy-Aware Data-Centric Generic Utility Based Approach in Wireless Sensor Networks

Wei-Peng Chen and Lui Sha

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{*wchen3,lrs*}*@cs.uiuc.edu*

Abstract. *Distinct from wireless ad hoc networks, wireless sensor networks are data-centric, application-oriented, collaborative, and energy-constrained in nature. In this paper, formulate the problem of data transport in sensor networks as an optimization problem whose objective function is to maximize the amount of information (utility) collected at sinks (subscribers), subject to the flow, energy and channel bandwidth constraints. Also, based on a Markov model extended from [3], we derive the link delay and the node capacity in both the single and multi-hop environments, and figure them in the problem formulation. We study three special cases under the problem formulation. In particular, we consider the energy-aware flow control problem, derive an energy aware flow control solution, and investigate via* ns-2 *simulation its performance. The simulation results show that the proposed energy-aware flow control solution can achieve high utility and low delay without congesting the network.*

1 Introduction

Recent technological advances have led to the emergence of small, low-power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities. Pervasive networks of such sensors and actuators open new vistas for constructing complex monitoring and control systems. Unlike traditional wired or wireless networks, sensor networks possess certain characteristics which warrant their treatment as a special class of ad hoc networks:

1. *Data-centric*: Sensor networks are largely data-centric, with the objective of delivering collected data, in a timely fashion, to destinations that require such data. Data that contains information of different qualities represents different values to destinations. As a result, the overall system objective is no longer to maximize the raw data throughput. Instead, maximizing the amount of useful information carried to destinations is an important criterion.
2. *Application-oriented*: While traditional wired and wireless networks are expected to cater to a variety of applications, sensor networks are usually deployed to perform specific tasks. The specific algorithms/protocols and performance metrics used in sensor networks thus depend on the characteristics and requirements of applications. For instance, for mission-critical applications, it is very important to ensure the end-to-end latency be kept below certain threshold.
3. *Collaborative*: Because of the application-oriented nature of sensor networks, how nodes collaborates with each other to realize the global system objective outweigh the objective of achieving fairness of individual connections. This is in sharp contrast to conventional wired and wireless networks in which provisioning of fairness to users is an important design criterion.
4. *Energy-constrained*: As most of the low-power devices in sensor networks have limited battery life and replacing batteries on tens of thousands of these devices is infeasible, any protocol/algorithm that will be eventually deployed in sensor networks has to be energy aware.

As a result of the unique characteristics of sensor networks, conventional routing and flow control protocols that focus on maximizing raw data throughput and achieving fairness are no longer well suited for sensor networks.

Instead, data-centric, utility based approaches that differentiate the treatments of packets with respect to their different values and at the same time, take into account of energy consumption are more adequate.

In this paper, we formulate the problem of data transport in sensor networks as an optimization problem whose objective function is to maximize the amount of information (utility) collected at sinks (subscribers), subject to the flow, energy and channel bandwidth constraints. In particular, we introduce energy constraints and the notion of quality of data into the formulation. Also, based on a Markov model extended from [3], we derive the link delay and the node capacity in both the single and multi-hop environments, and figure them in the problem formulation. As a result, the resulting solution approach will solve simultaneously the problems of maximizing utility and mitigating congestion.

We show that the formulated optimization problem is general enough to encompass a wide variety of applications in sensor networks, each with a different objective function and subject to different constraints. Specifically, we consider six design dimensions and adapt the generic formulation to meet the various needs of different applications. Following that we either modify the objective function or relax one of the constraint functions to conduct three special case studies under the generic problem formulation. In particular, we show in the first two case studies that the issue of routing in an environment monitoring system (which was considered in [5]) and the bandwidth allocation problem (which was considered in [14, 18]) can be treated as special cases under the generic problem formulation. In the third case study, we derive an energy aware flow control solution, and investigate via *ns-2* simulation its performance. The simulation results show that as compared with the Ad hoc On Demand Distance Vector (AODV) routing and load balancing routing, the solution derived under the proposed approach achieves higher utility and incurs lower latency. As will be elaborated on in Section 5, the simulation results also shed several insights that will be used to facilitate further improvement of utility-based approaches.

Utility based approaches have been explored in conventional wired networks (e.g., [10,12]), cellular wireless networks (e.g., [15]), ad hoc networks (e.g., [14, 18]), and most recently sensor networks [4]. Kelly *et al.* [10] propose a pricing scheme to achieve weight proportional fair rate allocation for users in the wireline environment. The same problem considered in [10] is solved by Low *et al.* [12] differently such that the dual problem can be optimized in a distributed manner. Both Xue *et al.* [18] and Qiu *et al.* [14] extended Kelly's work [10] and consider the rate allocation problem in ad hoc networks. The major differences lie in that (i) the former [18] uses the link capacity as the constraint of the channel capacity, while the latter [14] uses the node capacity as the constraint; and (ii) while the formulations in both the work reported in [18] and [10] divide the system problem into the user and network problems, the work reported in [14] incorporates the forwarding cost in the user optimization problem. None of the work in [10, 12, 18, 14] consider the energy constraints which we believe is one of he most important criteria in sensor networks. Saraydar *et al.* [15], on the other hand, take a utility based approach to control power of transmission in a decentralized manner in a multicell wireless data system. Recently Byers *et al.* [4] consider the optimization problem of maximizing the overall utility of sensor networks during the system lifetime, subject to an energy constraint that is expressed as a high level cost. Chang *et. al.* [5] devise a routing solution to maximize the system life time of sensor networks. As neither the link capacity nor the node capacity is considered in their work, the solution thus derived may not be feasible. Also, none of the existing work differentiates the treatment of packets with respect to their quality or information. In contrast, our proposed approach not only considers the energy constraint but also differentiates treatment of packets with respect to their quality of information.

The rest of the paper is organized as follows. We present the generic problem formulation in Sec. 2 and the three case studies in Sec. 3. Then, we derive in Sec. 4 the link delay and the node capacity that are necessary in the problem formulation. Following that, we present the simulation results for the third case study — the energy aware flow control problem — in Sec. 5. Finally we conclude the paper with a list of future research agendas in Section 6.

2 Problem Formulation

In this section we formulate a general utility-based optimization problem that can be tailored to fulfill various goals and requirements for different applications in sensor networks. Before delving into the problem formulation, we state the assumptions made in this paper:

(A1) Spatial redundancy is not considered: we assume that the sensing data collected from sensors at different locations contributes additive utilities. In reality, surplus sensors may be deployed in the sensing area and the information collected by neighboring sensors may be redundant and correlated. Clustering techniques such as GAF [17] or SPAN [6] have been proposed to group sensors into groups and coordinate activities among them, such that only one sensor needs to be awake in each group to maintain network connectivity and to carry out the sensing task. The data collected in different groups is likely non-redundant.

(A2) The utility of data packets originated from the same node is represented by a single utility function, in spite of the fact that they may be routed along different paths to the sinks.

(A3) The communication cost between sinks is negligible. Once data packets arrives at any of the sinks, they may be relayed to other sinks perhaps via a wireline network, and hence the communication cost is minimal.

The optimization problem is formulated to maximize the total utility of data collected at sinks throughout the system lifetime, subject to the flow constraint, the energy constraint and the channel capacity constraint. For notational convenience, we define the following notions:

- $\mathcal{U}_s(\cdot)$: the utility function that specifies the commodity generated from a sensor s and sent to a sink (perhaps through multiple routes);
- S_n and S_i: the set of sensors and sinks in the sensing field;
- N_k: the set of one-hop neighbors of node k;
- $q_{ij}^{(s)}$: the rate of the commodity s that passes from node i to node j;
- x_i: the source rate originated from node i;
- $x_i^{(s)}$: the source rate of commodity s originated from node i; As the commodity s only originates from node s, $x_i^{(s)} = x_i$ if $i = s$; otherwise, $x_i^{(s)} = 0$;
- E_i: the amount of energy initially equipped with node i;
- e_i: the energy consumed in the idle state per unit time;
- e_s and e_r: the additional energy consumed in transmitting and receiving one unit of data rate per unit time;
- $\overline{d_s}$: the average end-to-end latency that a packet experiences from a sensor s to a sink;
- T: the system lifetime defined as the time interval till the first failure of a node due to the depleted power;
- C_i: the channel capacity of node i; we will derive this value in Section 4.

Given the above notations, the problem can be formulated as a nonlinear programming problem as follows:

$$\max_{\mathbf{x},\mathbf{q},T} \Big[\sum_{s\in S_n} \mathcal{U}_s\Big(\sum_{i\in S_i} \sum_{k:i\in N_k} q_{ki}^{(s)}, \overline{d_s}\Big)\Big] \cdot T \tag{1}$$

$$s.t. \sum_{k:i\in N_k} q_{ki}^{(s)} + x_i \geq \sum_{j:j\in N_i} q_{ij}^{(s)}, \forall i \in S_n, s \in S_n \tag{2}$$

$$\Big\{e_r \cdot \sum_{k:i\in N_k} \sum_{s\in S_n} q_{ki}^{(s)} + e_s \cdot \sum_{j:j\in N_i} \sum_{s\in S_n} q_{ij}^{(s)} + e_i\Big\} \cdot T \leq E_i, \forall i \in S_n \tag{3}$$

$$\sum_{j:j\in N_i} \sum_{s\in S_n} q_{ij}^{(s)} \leq C_i, \forall i \in S_n \tag{4}$$

The objective is to maximize the utility of all received packets within the system lifetime over a vector of source rates of commodities ($\mathbf{x}$), a vector of link flow ($\mathbf{q}$) and the system lifetime (T). As such, the objective function (Eq. (1)) is expressed as the product of the system lifetime and the utility of all commodities received at the sinks per unit time. The utility function for the commodity s is a function of the total rate of the commodity s arriving

at sinks and the average end-to-end latency it sustains. By assumptions (**A2**) and (**A3**), the rate of commodity s arriving at sinks is the sum of all the incoming flows of commodity s to any of sinks. Since flows traveling through different routes to sinks endure different latencies, we express the utility function as a function of the average latency $\overline{d_s}$ to account for the average loss of utility due to the delay. Moreover, with different qualities of data, the different quantized utility functions (such as in [11]) can be used to evaluate the utility of a data packet.

The first constraint (Eq. (2)) is the flow conservation. The sum of both the incoming flows of commodity s and the flow of commodity s originated from a node is greater than or equal to the sum of the outgoing flows of commodity s, with the inequality implying that intermediate relay nodes may drop packets they forward. The second constraint (Eq. (3)) is the energy constraint, while the third constraint (Eq. (4)) is the capacity constraint, i.e., the sum of the outgoing flows of all the commodities from a node i should be less than its channel capacity C_i (the value of which will be derived in Section 4).

The problem formulated above aims to maximize the total utility received at the sinks, by controlling the parameter vectors $\mathbf{x}$ and $\mathbf{q}$ (which in turn are related to flow control and routing decisions). As a matter of fact, the above problem formulation encompasses a wide variety of requirements and objectives for different applications in sensor networks. In what follows, we discuss six possible design dimensions and their corresponding amendments to the above problem formulation:

1. *Flow conservation*: If intermediate relay nodes are not allowed to discard packets they forward, the inequality in Eq. (2) is changed to an equality. With the flow conservation constraint, for each commodity s, the sum of the incoming flows of commodity s at sinks is equal to the rate x_s originated at node s, and hence the objective function in Eq. (1) can be rewritten as:
$$\max_{\mathbf{x},\mathbf{q},T} [\sum_{s\in S_n} \mathcal{U}_s(x_s, \overline{d_s})] \cdot T \tag{5}$$
2. *Flow indivisibility constraint*: If a commodity from a sensor node s cannot be routed through multiple paths, an additional constraint has to be added in Eq. (2) such that for each commodity s, only one incoming and one outgoing flow has positive rate and others are zero. This makes it more difficult to solve the optimization problem because of its discrete constraint.
3. *Flow control*: In the problem formulation, both the routing and flow control problems are jointly considered. An alternative approach is to solve the optimization problem in two steps. The routes are determined first by a routing protocol and figured into the optimization problem. The optimization problem then solves the flow control problem, by optimizing the total utility over the vector of source rates, $\mathbf{x}$.
4. *Quality-driven utility function*: If the quality of data is considered, the utility function of each sensor is determined based on the quality of the data sensed; otherwise, the utility functions are the same for all the sensors.
5. *Effect of latency on utility*: Whether the latency affects the utility of the data sensed is application-dependent. In general, the utility of data decays with the latency but the decay function (convex, linear, or concave) varies with the application characteristics.
6. *Energy awareness*: If the energy constraint is not considered, the problem formulation can be simplified as follows: The system lifetime T can be removed from the objective function and the energy constraint (Eq. (3)) can be removed.

Whether or not a utility-based approach is effective is contingent upon the proper design of the utility function. Several rules can be applied to determine appropriate utility functions in sensor networks: (i) More data generates more utility, but the marginal utility decreases due to spatial and temporal redundancy; (ii) The utility of data decreases with the latency; (iii) Better data quality results in higher utility. Based on the above rules, we characterize the utility functions as follows:
$$\mathcal{U}_s(x_s, \overline{d_s}) = a_s \cdot \mathcal{U}(x_s), \cdot \mathcal{D}_s(\overline{d_s}) \tag{6}$$
where a_s is a parameter that determines the importance of, and the quality of, data originated from sensor s. The function $\mathcal{U}(\cdot)$ is a non-decreasing, concave utility function of the source rate. The function $\mathcal{D}_s : \mathcal{R}^+ \rightarrow [0, 1]$

is a non-increasing utility decay function of the average end-to-end latency, $\overline{d_s}$, of packets from sensor s. $\mathcal{D}_s(\cdot)$ is application-dependent, and can be convex, linear, concave functions or a combination thereof. For mission-critical applications, the value of utility decays abruptly after the system tolerance period and is hence a convex function in that region. On the other hand, for non-mission-critical applications, the utility is not significantly affected by the latency and can thus be expresses as a concave function. The average end-to-end latency of packets from sensor i can be estimated as

$$\overline{d_s} = \sum_{j,k} q_{jk}^{(s)} \cdot d_{l_j} / \sum_{j,k} q_{jk}^{(s)}, \tag{7}$$

where d_{l_j} is the link delay from node j to its next hop and will be derived in Sec. 4. For simplicity, we only consider the accumulated link delay in Eq. (7), but ignores the queuing delay. With the capacity constraint in Eq. (4), we expect the queuing delay should not be significant.

3 Application Examples Under the Problem Formulation

In this section, we consider three representative problems that have been considered in the literature, and show that they are special cases of the general problem formulation given in Section 2.

3.1 Case Study I: Routing in a Environment Monitoring System

Consider an environment monitoring system in which sensor nodes monitor environmental changes such as temperature, moisture, habitat, chemical contaminant or construction safety. Sensors that are deployed in an area periodically send their sensor readings back to the control center, so that data can be logged and/or further analyzed. The sending rate $\mathbf{x}$ is usually given, and the utility of data from all sensors is the same and does not decay with the latency. As the objective of such a system is to maximize the system lifetime under the condition that the sensors cover the entire area, the objective function (Eq. (1)) is modified as $\{\max_{\mathbf{q}} T\}$, with the constraints in Eqs. (2)–(4) remaining the same.

This formulation is similar to that in [5], with a major difference: the formulation in [5] only considers the transmission power, but not the node capacity constraint. Without considering the node capacity constraint, the problem was reduced to a linear programming problem (that may render an infeasible solution).

3.2 Case Study II: Flow Control

The problem formulated in Section 3.1 assumes that the source rate of commodities $\mathbf{x}$ has been given and considers the routing problem by determining $\mathbf{q}$ to maximize the system lifetime. On the other extreme, one can assume that the routes are given for each pair of source and destination and maximizes the utility of received packets at sinks by controlling the vector of source rate $\mathbf{x}$. This is termed as the flow control problem (a.k.a. the bandwidth allocation problem). Several efforts have been made along this research avenue: Kelly *et al.* [10] proposed a price-based bandwidth allocation approach to maximize the total utility of all users in wireline environments. Xue *et al.* [18] applied the same formulation to ad hoc networks. In both work only the constraint on the link capacity is considered. Note that although considering the constraint on the link capacity is adequate in wireline environments, it may not be sufficient in ad hoc environments. Instead the constraint on the node capacity should be considered, as all outgoing links from a node share the same channel. The constraint on the link capacity suffices only when a node is equipped with multiple transceivers operating at different channels or with directional antennas. Qiu *et al.* [14] considered the constraint on the node capacity and further relaxed the flow conservation rule.

All the above work does not consider the energy constraint and thus can only be applied to sensors that come with tethered power supplies. Without the energy constraint, the system lifetime, T, is no longer a control variable and the objective function becomes separable, rendering a separable nonlinear programming problem. Note that the dual problem of a separable nonlinear programming problem can be readily evaluated [2], and a distributed solution can also be derived [12].

1. $G \leftarrow S_i$
2. $Q \leftarrow S_n$ /*sorted in the increasing order of distance to any one of sinks*/
3. while $(Q \neq \emptyset)$
4. $\quad s \leftarrow Q.head()$; $s.cost \leftarrow 0$
5. $\quad$ Run Dijkstra's algorithm to find the shortest path p from s to its closest sink using nodes $\in G$
6. $\quad$ Backtrack the path p from the sink to s and $\forall$ nodes $v \in p \backslash S_i$, $v.cost \leftarrow v.cost + 1$
7. $\quad G \leftarrow G \bigcap \{s\}$

Fig. 1. The routing algorithm that balances loads based on geographical information.

3.3 Case Study III: Energy-Aware Flow Control

In sensor networks, energy saving is the top priority for system design. Hence, the problem formulation in Section 3.2 can be augmented by adding the energy constraint and figuring in the system lifetime in the objective function (as in Eq. (5)). Recall that the optimization problem in Eq. (5) considers both routing and flow control decisions together, and is a NP-hard problem as shown in [20]. A simple solution of the augmented problem can be derived into two phases. In the first phase, based on the geographical information, a load balancing route is first determined for each sensor using the algorithm given in Fig. 1. The algorithm rests on the assumption that a link $l_{i \rightarrow j}$ is in G if the distance between i and j is less than the radio range and j is closer to the destination than i (greedy geographical routing). It finds routes for sensors, starting from the sensor that is closest to any of the sinks. The cost associated with the node represents the number of flows it originates or forwards for other nodes. Since a route is assigned for each sensor sequentially, when a node further away from a sink determines its route, it attempts not to select nodes with high costs. Therefore, the objective of load balance can be achieved, and this routing algorithm serves as a good basis for finding a good solution for the optimization problem. The resulting routes derived in the algorithm are expressed in the routing matrix A, with $A_{sf} = 1$ if the node f is on the route of sensor s to its closest sink; and $A_{sf} = 0$ otherwise.

In the second phase, given the set of routes for all the sensors and under the assumption that the utility of data does not decay with the latency, the optimization problem becomes (let I and $\mathbf{e}$ be identity matrix and entity vector, respectively):

$$\begin{aligned} \max_{\mathbf{x},T} \; & [\sum_{s \in S_n} \mathcal{U}_s(x_s)] \cdot T \\ s.t. \; & [e_s \cdot A^T \cdot \mathbf{x} + e_r \cdot (A^T - I) \cdot \mathbf{x} + e_i \cdot \mathbf{e}] \cdot T \leq \mathbf{E}, \\ & A^T \cdot \mathbf{x} \leq \mathbf{C}. \end{aligned} \tag{8}$$

Recall that $\mathbf{x}$, $\mathbf{E}$, and $\mathbf{C}$ are the vectors of source rate, initial energy, and node capacity for sensors, respectively. The vector $\mathbf{x}$ is the control variable, $\mathbf{E}$ is given by the system, and $\mathbf{C}$ is a topology-dependent parameter and will be derived in Section 4.

Unfortunately, both the objective function and the first constraint are not convex functions. As a result, a solution satisfying Karush-Kuhn-Tucker conditions may not be the global optimal solution. However, the above solution gives a lower bound for the optimal solution, and can be used as an approximate solution. Several global optimization methods without searching the global optimal solution exhaustively have been proposed. Vanderbei et. al. [16] propose an efficient approach for non-convex nonlinear programming by slightly modifying the interior-point approach for quadratic programming. The interested reader is referred to [7] for several global optimization methods. In this paper, we use $MATLAB^{\textcircled{R}}$ to find the solution by attempting on several random initial points. The performance of the solution under this problem formulation will be evaluate in Section 5.

4 Derivation of Link Delay and Node Capacity in 802.11-like MAC Protocol

In the general optimization problem (Eqs. (1)-(4)), the average end-to-end latency is needed in Eq. (1) and the node capacity is needed in Eq. (4). Recall that by Eq. (7), we have simplified the end-to-end latency as the accumulated

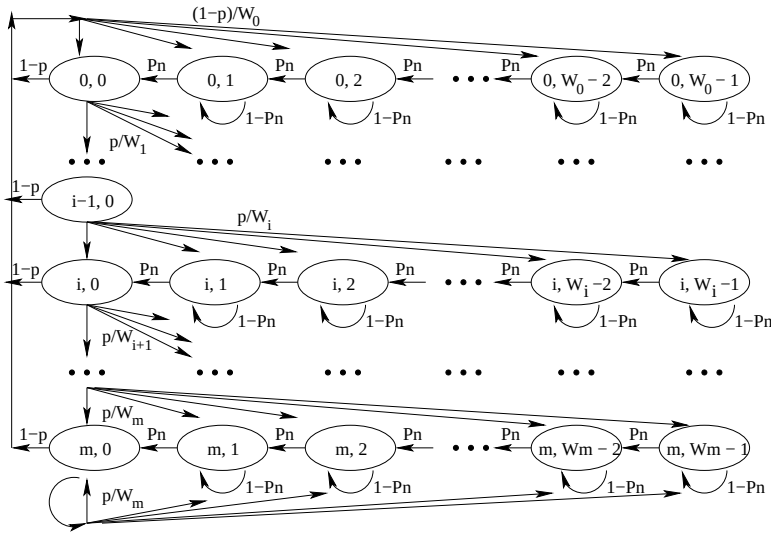


Fig. 2. The Markov chain model for analyzing the link delay under an IEEE 802.11-like backoff mechanism.

link delay. In this section, we derive both parameters, assuming that an IEEE 802.11-like backoff mechanism has been used as the underlying MAC protocol.

In the 802.11-like backoff mechanism, before a node first attempts to send a packet, it sets a backoff timer uniformly distributed in $[0, W_0]$, where $W_0 = W$ is the initial maximum contention window. A node counts down its backoff timer whenever the channel is sensed idle; otherwise, it freezes the timer. When the value of the backoff timer becomes zero, the node transmits the packet. If the packet collides with other packets, i.e. the sender does not receive an acknowledgment, the backoff timer is reset to be uniformly distributed in $[0, W_i]$, where the index i represents the number of retransmissions, and $W_i = 2^i \cdot W_0 = 2^i \cdot W$ if the exponential binary backoff policy is adopted. The *maximum backoff state* is m, i.e., the maximum window size $W_{max} = 2^m \cdot W$.

4.1 Derivation of Link Delay in the Single Hop Case

The derivation of the link delay within one hop is based on the model in [3]. Two major assumptions (approximations) are made in our model (and also in [3]). First, the network consists of M backlogged nodes within the one-hop transmission range of each other. That is, each node in the system can hear each other and always has packets to send. Second, the time after detecting an idle channel is slotted, and in each transmission attempt the probability, p, that a packet collides with some other(s) in a slot is a constant, regardless of the number of retransmissions the packet has incurred. The probability p is referred to as *conditional collision probability*. The model can be described with a Markov chain shown in Fig. 2, with the state being a 2-tuples $(s(t), b(t))$, where $s(t)$ is the retransmission times (ranging from 0 to m, the maximum backoff state) and $b(t)$ is the value of the backoff timer of a node at time t.

The major difference between our model and that in [3] lies in that the later does not consider the effect of freezing the backoff timer when a node in its backoff period senses the channel busy. To account for this effect, there are two state transitions from state (i, j) $(j > 0)$ to other states in our model: (i) one is with the probability of sensing the channel idle, p_n, i.e., the other $M - 1$ nodes do not send packets, and the state becomes $(i, j - 1)$ as a result of the backoff timer being decremented by one; and (ii) the other one is with the probability of sensing the channel busy, $1 - p_n$, i.e., at least one other node sends a packet in the slot, and the state remains at (i, j). Note that p_n is the conditional probability of sensing the channel idle given that the node itself does not send any packet (note that $j > 0$) in the current time slot, and hence $p_n = (1 - \tau)^{M-1}$, where τ is the probability that a node transmits a packet in a slot time.

By the operations of the backoff mechanism, the state transition probabilities can be expressed as follows:

$$\begin{cases} P\{i,k|i,k\} = 1-p_n, & k \in (1, W_i-1) \;\; i \in (0,m), \\ P\{i,k-1|i,k\} = p_n, & k \in (1, W_i-1) \;\; i \in (0,m), \\ P\{0,k|i,0\} = (1-p)/W_0, & k \in (0, W_0-1) \;\; i \in (0,m), \\ P\{i,k|i-1,0\} = p/W_i, & k \in (0, W_i-1) \;\; i \in (1,m), \\ P\{m,k|m,0\} = p/W_m, & k \in (0, W_m-1). \end{cases} \tag{9}$$

Let $b_{i,k} = \lim_{t\to\infty} P\{s(t)=i, b(t)=k\}, i \in (0,m), k \in (0, W_i-1)$ be the stationary distribution of state (i,k). Given the Markov model in Fig. 2 and the state transition probabilities in Eq. (9), the equilibrium equations for the states $(i, W_i-1), (i,k), 0<k<W_i-1$, and $(i,0)$ in the case that the number of retransmissions is $i, 0<i<m$ are:

$$\begin{cases} b_{i,W_i-1} = \frac{p}{W_i}\cdot b_{i-1,0} + (1-p_n)\cdot b_{i,W_i-1} & \Longrightarrow b_{i,W_i-1} = \frac{p}{W_i\cdot p_n}\cdot b_{i-1,0}, \\ b_{i,k} = p_n\cdot b_{i,k+1} + \frac{p}{W_i}\cdot b_{i-1,0} + (1-p_n)\cdot b_{i,k} & \Longrightarrow b_{i,k} = b_{i,k+1} + \frac{p}{W_i\cdot p_n}\cdot b_{i-1,0} = \frac{(W_i-k)p}{W_i\cdot p_n}\cdot b_{i-1,0}, \\ b_{i,0} = \frac{p}{W_i}\cdot b_{i-1,0} + p_n\cdot b_{i,1} = p\cdot b_{i-1,0} = p^i\cdot b_{0,0}. & \end{cases} \tag{10}$$

Similarly, the equilibrium equations for the states $(m, W_m-1), (m,k), 0<k<W_m-1$, and $(m,0)$ in the case that the number of retransmissions is equal to or more than the maximum backoff stage m become:

$$\begin{cases} b_{m,W_m-1} = \frac{p}{W_m}(b_{m-1,0}+b_{m,0}) + (1-p_n)b_{m,W_m-1} & \Longrightarrow b_{m,W_m-1} = \frac{p}{W_m\cdot p_n}(b_{m-1,0}+b_{m,0}), \\ b_{m,k} = \frac{p}{W_m}(b_{m-1,0}+b_{m,0}) + p_n\cdot b_{m,k+1} + (1-p_n)b_{m,k} & \Longrightarrow b_{m,k} = \frac{(W_m-k)p}{W_m\cdot p_n}(b_{m-1,0}+b_{m,0}), \\ b_{m,0} = \frac{p}{W_m}(b_{m-1,0}+b_{m,0}) + p_n\cdot b_{m,1} & \Longrightarrow b_{m,0} = \frac{p}{1-p}\cdot b_{m-1,0} = \frac{p^m}{1-p}\cdot b_{0,0}. \end{cases} \tag{11}$$

Finally, the equilibrium equations for the states $(0, W_0-1), (0,k), 0<k<W_0-1$, and $(0,0)$ in the case that the packet being sent incurs no retransmission are:

$$\begin{cases} b_{0,W_0-1} = (1-p_n)b_{0,W_0-1} + \frac{1-p}{W_0}\cdot\sum_{i=0}^{m} b_{i,0} & \Longrightarrow b_{0,W_0-1} = \frac{1-p}{W_0\cdot p_n}\cdot\sum_{i=0}^{m} b_{i,0}, \\ b_{0,k} = (1-p_n)\cdot b_{0,k} + p_n\cdot b_{0,k+1} + \frac{1-p}{W_0}\cdot\sum_{i=0}^{m} b_{i,0} & \Longrightarrow b_{0,k} = \frac{(1-p)\cdot(W_0-k)}{W_0\cdot p_n}\cdot\sum_{i=0}^{m} b_{i,0}, \\ b_{0,0} = p_n\cdot b_{0,1} + \frac{1-p}{W_0}\cdot\sum_{i=0}^{m} b_{i,0} & \Longrightarrow b_{0,0} = (1-p)\cdot\sum_{i=0}^{m} b_{i,0}. \end{cases} \tag{12}$$

The stationary probabilities of state $(i,0)$ and (i,k) in Eqs. (10)-(12) can be expressed in terms of $b_{0,0}$ as:

$$b_{i,0} = b_{0,0}\cdot\begin{cases} p^i, & 0<i<m, \\ \frac{p^m}{1-p}, & i=m, \end{cases} \tag{13}$$

and

$$b_{i,k} = \frac{(W_i-k)}{W_i\cdot p_n}\cdot b_{0,0}\cdot\begin{cases} 1, & i=0, \quad 0<k<W_0, \\ p^i, & 0<i<m, \; 0<k<W_i, \\ \frac{p^m}{1-p}, & i=m, \quad 0<k<W_m. \end{cases} \tag{14}$$

$b_{0,0}$ can be derived by equating the sum of the probabilities of all states to one, i.e.,

$$1 = \sum_{i=0}^{m}\sum_{k=0}^{W_i-1} b_{i,k} \Longrightarrow b_{0,0} = \frac{2p_n(1-p)(1-2p)}{(2p_n-1)(1-2p)+W(1-p-2^m p^{m+1})}. \tag{15}$$

Since a node only attempts to send a packet when its backoff timer is zero, we can express the probability τ that a node transmits a packet in a randomly chosen slot as the sum of state probabilities with $b(t)=0$:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1-p} = \frac{2p_n(1-2p)}{(2p_n-1)(1-2p)+W(1-p-2^m p^{m+1})}. \tag{16}$$

Note that τ is expressed in terms of the conditional collision probability, p. On the other hand, the conditional collision probability p is the probability that a packet being sent collides with packets from other nodes. That is, p can be expressed in terms of transmission probability τ as follows:

$$p = 1-(1-\tau)^{M-1} = 1-p_n. \tag{17}$$

With Eqs. (16) and (17) we may numerically solve both τ and p. The average number of attempts to transmit a packet can be expressed in terms of p:

$$\text{Ave. \# of attempts} = \sum_{i=0}^{\infty} i(1-p)p^{i-1} = \frac{1}{1-p}. \tag{18}$$

With all the state probabilities derived, we now derive the link delay — the expected latency to transmit a packet to the next hop. Let $d_{i,k}$ be the expected time[1] to transmit a packet when a node is at the state (i,k). By the Markov model given in Fig. 2, there are two state transitions from state $(i,k), k \neq 0$. Either the channel is idle or busy in the next slot. In the latter case, the node remains at state (i,k) for a time interval, $\overline{T_f}$, that is equal to the conditional expected freeze time given that the channel is busy.

$$d_{i,k} = p_n \cdot (1 + d_{i,k-1}) + (1-p_n) \cdot (\overline{T_f} + d_{i,k}) \Longrightarrow d_{i,k} = k \cdot [1 + \frac{(1-p_n) \cdot \overline{T_f}}{p_n}] + d_{i,0}, \quad \forall 0 \leq i \leq m, 0 < k < W_i \tag{19}$$

The conditional expected freeze time $\overline{T_f}$ can be expressed as

$$\overline{T_f} = \frac{(M-1)\tau(1-\tau)^{M-2}}{1-p_n} \cdot T_s + \frac{1-(1-\tau)^{M-1} - (M-1)\tau(1-\tau)^{M-2}}{1-p_n} \cdot T_c, \tag{20}$$

where the first term represents the case when only one node out of the other $M-1$ nodes sends a packet, the packet is successfully transmitted, and the channel is occupied for an interval of T_s, i.e., the expected time to successfully transmit a packet; and the second term represents the case when more than one node attempt to send packets, the packets collide with each other and the channel is busy for an interval of T_c, i.e., the expected collision period sensed by a node.

The values of T_s and T_c vary depending on whether or not the RTS/CTS mechanism is used. In the basic mode (denoted as bas), the RTS/CTS mechanism is not used, T_s^{bas} includes the packet header (H), the expected packet payload ($E[P]$), short inter-frame space ($SIFS$), acknowledgment (ACK), distributed inter-frame space ($DIFS$) and twice of the propagation delay (δ); and T_c^{bas} contains the packet header (H), the expected length of the longest packet payload involved in a collision ($E[P^*]$), $DIFS$, and δ. In this paper we assume all packets have the same payload size, and thus $E[P] = E[P^*] = P$. T_s^{bas} and T_c^{bas} are expressed as:

$$\begin{cases} T_s^{bas} = H + E[P] + SIFS + \delta + ACK + DIFS + \delta, \\ T_c^{bas} = H + E[P^*] + DIFS + \delta. \end{cases} \tag{21}$$

In the RTS/CTS mode (denoted as rts), the service time, T_s^{rts}, contains several additional terms, i.e., request-to-send packet, (RTS), clear-to-send packet (CTS), $2 \times SIFS$, and $2 \times \delta$, as compared with the basic mode. On the other hand, T_c^{rts} contains only RTS, $DIFS$, and δ.

$$\begin{cases} T_s^{rts} = RTS + SIFS + \delta + CTS + SIFS + \delta + H + E[P] + SIFS + \delta + ACK + DIFS + \delta, \\ T_c^{rts} = RTS + DIFS + \delta. \end{cases} \tag{22}$$

By the Markov model, state transitions of state $(i,0)$ are different from those of state $(i,k), k \neq 0$. At state $(i,0)$, a node attempts to send a packet, and incurs either a successful transmission (that takes time T_s), or a collision (that causes the channel to be busy for an interval of T_c and the backoff stage to increase by one). The expected time to transmit a packet for state $(m,0)$ and state $(i,0), 0 \leq i < m$ are respectively

$$\begin{cases} d_{m,0} = (1-p) \cdot T_s + \frac{p}{W_m} \sum_{k=0}^{W_m - 1} (d_{m,k} + T_c) & \Longrightarrow d_{m,0} = T_s + \frac{p}{1-p} \cdot T_c + \frac{p(2^m W - 1)}{2(1-p)} \cdot [1 + \frac{(1-p_n)\overline{T_f}}{p_n}], \\ d_{i,0} = (1-p) \cdot T_s + \frac{p}{W_{i+1}} \sum_{k=0}^{W_{i+1} - 1} (d_{i+1,k} + T_c) & \forall\, 0 \leq i \leq m-1. \end{cases} \tag{23}$$

Combining Eq. (19) and (23) and using induction from $m-1$ to 0, we can express $d_{i,0}, 0 \leq i < m$ as follows:

$$d_{i,0} = T_s + \frac{p}{1-p} \cdot T_c + \frac{p}{2(1-p)} \cdot [1 + \frac{(1-p_n)\overline{T_f}}{p_n}] \cdot [\frac{2^{i+1} pW(1-(2p)^{m-i-1})}{1-2p} + 2^{i+1}W - 1], \forall 0 \leq i < m. \tag{24}$$

[1] The unit of $d_{i,k}$ is the slot time.

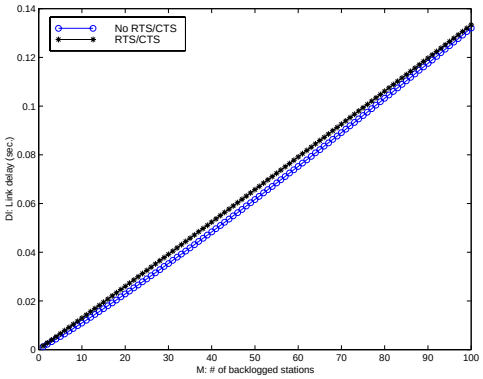


Fig. 3. The link delay in the basic mode and in the RTS/CTS mode in the one-hop environment.

Since the initial state is among the state $(0,0)$ to $(0,W-1)$, the average link delay D_l is the average time to transmit a packet with the average taken over state $(0,0)$ to state $(0,W-1)$. Thus,

$$D_l = \frac{1}{W} \cdot \sum_{k=0}^{W-1} d_{0,k} = T_s + \frac{p}{1-p} \cdot T_c + \frac{1 + \frac{(1-p_n)\overline{T_f}}{p_n}}{2(1-p)} \cdot (\frac{pW(1-(2p)^m)}{1-2p} + W - 1). \tag{25}$$

After several arithmetic operations, D_l can be expressed in terms of M and τ as follows:

$$D_l = M(T_s - T_c) + \frac{1-\tau}{\tau} \cdot \{1 + [(1-\tau)^{-M} - 1] \cdot T_c\}. \tag{26}$$

Although τ is also a function of M, it is not easy to express τ in terms of M. Instead both τ and p are numerically solved through Eq. (16) and Eq. (17). The numerical values of D_l under both the basic and RTS/CTS modes are given in Fig. 3. Surprisingly the curve of D_l is very close to a linear function of M. This is perhaps attributed in part by the fact that the first term $M(T_s - T_c)$ in Eq. (26) dominates the second term. The result is intuitively correct since 802.11 is fair for all backlogged nodes, and each such node takes turns to transmit a packet. As a result the one-hop link delay increases linearly with M. The numerical results of D_l in the single hop case will serve a base for deriving the link delay under the multiple-hop environment.

4.2 Derivation of Link Delay in the Multiple Hop Case

The major hurdle in deriving the link delay in the multiple hop case is that the assumption that all nodes can hear each one another no longer holds. The transmission of a node may interfere with that of another node outside its radio range. This is know as the *hidden terminal problem*. In general, the number of backlogged nodes that compete with a node depends on the locations and the traffic loads of other nodes. The flow contention graph proposed in [13, 18, 9] is commonly used to approximate the number of backlogged nodes with which a node may contend. In the flow contention graph, a vertex and an edge represent, respectively, a flow and the potential contention between two flows. For example, as shown in Fig. 4 flow $F1$ may interfere with flows $F2$, $F3$, $F5$, and $F6$ and thus the vertex $F1$ is adjacent to those vertices in flow contention graph.

To infer the exact number of backlogged nodes that compete with a sender node s, node s needs to know the buffer status, the link traffic and the scheduling algorithm at other nodes, which is essentially infeasible. Instead, we observe that two flows contend with each other if either the sender or the receiver of one flow is within the transmission range of either the sender or receiver of the other flow, and approximate the number of potentially competing backlogged nodes as the number of neighbors that lie within the two-hop neighborhood of node s. That is, we estimate the link delay incurred by a flow emanating from node s as $D_l(|N_s|)$, where $|N_s|$ is the number of neighbors within node s's two hops neighborhood. Note that this approximation over-estimates the number of potentially conflicting backlogged nodes and we err on the pessimistic side.

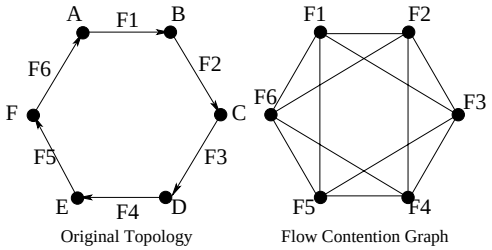


Fig. 4. An example that shows how the flow contention graph is derived from the original topology. This figure is reprinted from [13].

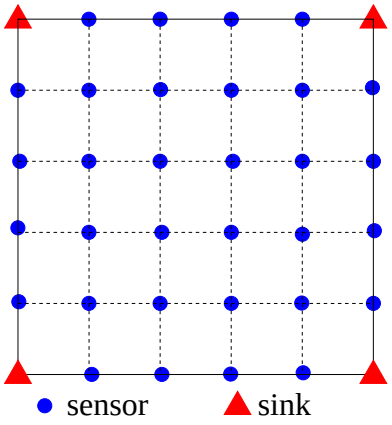


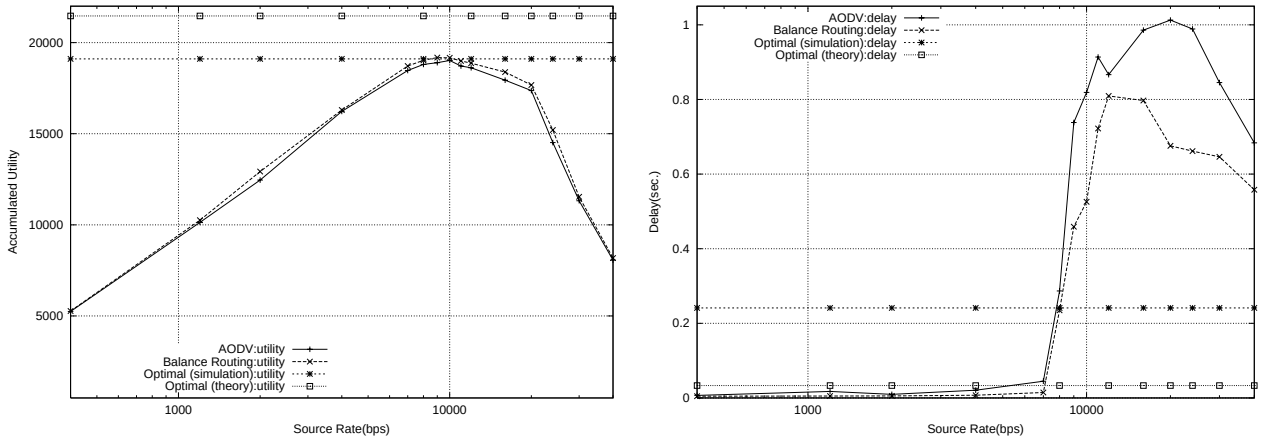
Fig. 5. The topology used in the simulation study. A total of 4 sinks and 32 sensors are deployed on a square grid. The distance between neighboring sensors is 200 meters, and each sensor has at most 4 neighbors.

4.3 Derivation of Node Channel Capacity

After deriving the link delay, we now derive the node capacity. Since all the flows emanating from node s share a single interface, the throughput attained by these flows should be less than node s's channel capacity C_s. Hence we estimate node s's channel capacity C_s as the reciprocity of the link delay $1/D_{l_s}$. The link delays D_l is used to estimate the end-to-end delay in the utility function Eq. (1), while the node capacity C is used in Eq. (4) to constrain the total commodity rate from a node.

5 Performance Evaluation of Energy Aware Flow Control Problem

In this section, we evaluate via *ns-2* simulation the performance of the energy-aware flow control problem (case III) formulated in Sec. 3. We use the topology depicted in Fig. 5 in the simulation: 4 sinks and 32 sensors are deployed on a square grid. The distance between neighboring sensors is 200 meters. Since the default transmission range in IEEE 802.11 is 250 meters, each sensor has at most 4 neighbors. The optimization problem to be solved is given in Eq. (8) in which the routing matrix A is determined by the algorithm given in Fig. 1, and the utility function is defined as $\mathcal{U}_s(x_s) = v_s \cdot log(x_s + 1)$, where v_s and x_s are the utility value of a packet and the source rate (in units of #packets/second) for sensor node s, respectively. Function $\mathcal{U}_s$ is a non-decreasing and concave function of node s's sending rate. The energy, E_i, each sensor is initially equipped with is 200 joules. The parameters for energy consumption follows the setting in [6], i.e., the energy consumption incurred in the transmission, reception, and idle state is 1.4, 1.0, and 0.83 W, respectively. Hence, e_i is 0.83 W. e_s and e_r are the additional energy consumed (in addition to e_i) in sending and receiving a packet, and are equal to the product of T_s, the time to send a packet and 0.57(=1.4-0.83) and 0.17(=1.0-0.83), respectively. (Note that the units of e_s and e_r are joules per packet.) The



(a) (b)

Fig. 6. Performance comparison of the energy-aware flow control solution against AODV routing and load balanced routing with respect to (a) utility and (b) end-to-end packet delay under a wide range of source rates. The utility values of packets are the same.

payload size of a packet is set to be 70 bytes (including 20 bytes of IP header but not MAC and PHY headers). With the above parameter setting, the source rate, x_i of each sensor i is obtain by solving Eq. (8) using $MATLAB^{\textcircled{R}}$.

Comparison with respect to accumulated utility and end-to-end delay: We carry out two sets of simulations. In the first set of simulations, the utility values of packets from all the sensors are assumed to be equal to 1 while in the second set of simulations, the utility value of packets from node s, v_s, is uniformly distributed in $[1, 100]$. We compare the solution derived under the energy-aware flow control problem with two routing approaches, AODV [1] and load balanced routing in Fig. 1 under a wide range of source rates. Fig. 6 gives the accumulated utility and the end-to-end packet delay under the optimal energy-aware flow control, AODV routing, and load balanced routing in the first set of simulations. The performance of load balanced routing is slightly better than that of AODV with respect to both the accumulated utility and the end-to-end delay. This is in part due to the fact that load balanced routing is based on static information and does not incur routing overhead. Both AODV and balanced routing achieve the highest utility at the source rate of approximately 9 to 10 kbps. The utilities decrease as the source rate deviates from the optimal point. Moreover, the best results achieved by both AODV and balanced routing are almost the same as the derived solution in which the source rate of 8 sensors one hop away from sinks is 8.57 kbps and the source rate of other sensors is roughly 7.95 kbps. (Note that because the utility values of packets for all the sensors are the same, the optimal source rate is almost the same for all sensors.) The end-to-end delay, on the other hand, increases dramatically under AODV and balanced routing when the source rate exceeds 7 kbps (beyond which the network capacity has saturated, Fig. 6(b)).

Fig. 7 gives the accumulated utility and the end-to-end packet delay under the optimal energy-aware flow control, AODV routing, and load balanced routing in the second set of simulations. The results exhibit similar trends as those in the first set of simulation. The highest utility under AODV and load balanced routing occurs when the source rate is approximately 9 kbps, and the end-to-end delay increases dramatically when the source rate exceeds 7 kbps. The major notable difference is that the flow control solution achieves higher utility than balanced routing by 4%, due to the use of different utility values for packets that originate from different sensors. Also, in the energy-aware flow control solution the difference in the source rates for sensors with different values is significant. The reason why the performance improvement is not significant is two-fold: First, the choice of logarithmic utility functions achieves high utility quickly but the gain beyond further increasing the source rate is marginal. As a result, applying high enough source rates (but not so high as to severely congest the network) to all the sensors (and hence necessarily to the sensors with high utility values) can thus achieve high utility. Second, the energy

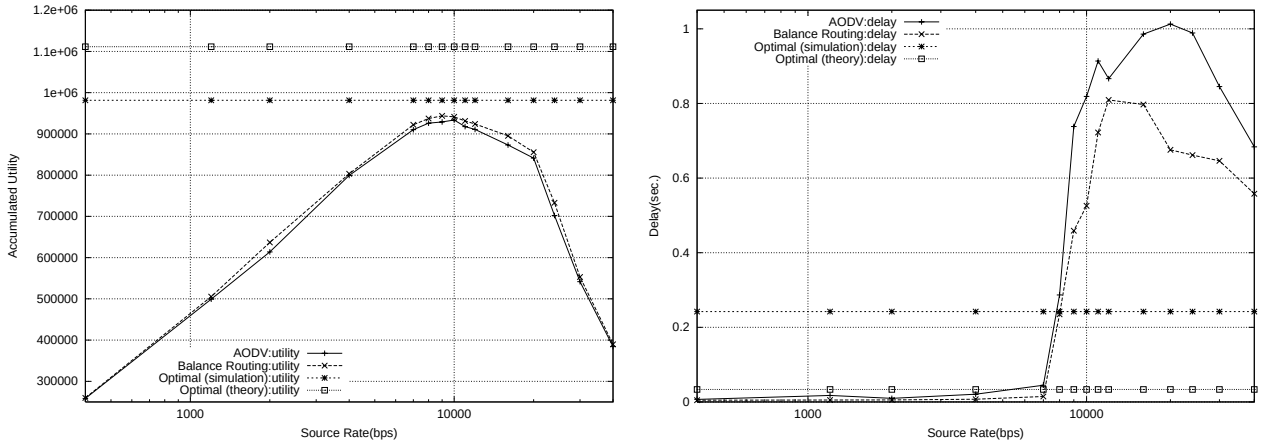


Fig. 7. Performance comparison of the energy-aware flow control solution against AODV routing and load balanced routing with respect to (a) utility and (b) end-to-end packet delay under a wide range of source rates. The utility values of packets are uniformly distributed in $[1, 100]$.

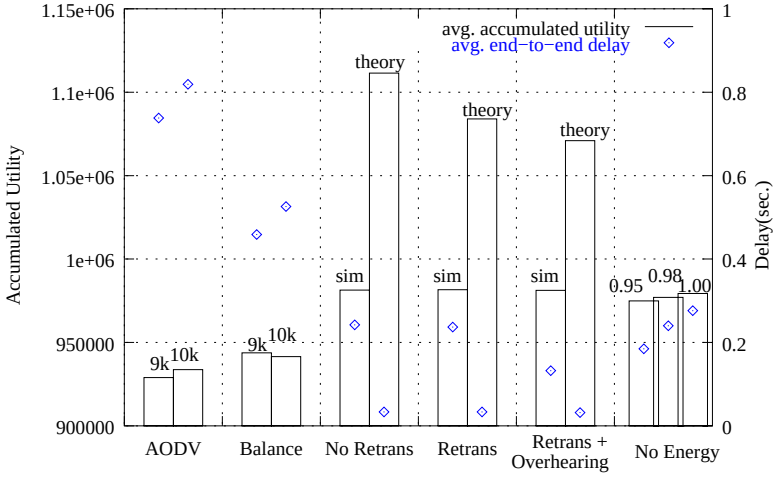


Fig. 8. Compensated results of optimal flow control and best results of AODV and load balanced routing. "Retrans" denotes the results obtained with the retransmission effect compensated, and "Retrans + Overhearing" denotes the results obtained with both the retransmission and overhearing effects compensated.

consumed in the idle state dominates the total energy consumed because e_i is only slightly smaller than e_s and e_r and the period of idle state is much longer than that of transmitting or receiving state. The insignificant energy additionally incurred in transmitting or receiving packets justifies for large source rates. In other words, the energy constraint is not as stringent as compared to the node capacity constraint. We will later elaborate on this point by comparing the energy-aware flow control solution obtained without the energy constraint.

There exists a gap between the solution obtained in the simulation and that obtained numerically through the theoretical result with respect to both the utility and the end-to-end delay. This may be a result of inaccurate modeling of the energy constraint in two aspects. First the values of e_s and e_r in Eq. (3) ignore the energy consumed in packet retransmissions. Second, the energy incurred in receiving packets in Eq. (3) only takes into account of received packets destined for itself but not overheard packets. To compensate for the first effect, we apply the retransmission factor obtained from Eq. (18) to both e_s and e_r. Similarly to compensate for the second effect, we

Routing	AODV		Balance		Optimal(simulation)			Optimal(theory)		
Variations	9kbps	10kbps	9kbps	10kbps	no retrans	retrans	overhearing	no retrans	retrans	overhearing
Utility	928948	933702	943762	941475	981404	981605	981244	1111443	1083962	1070974
Avg. Delay(sec)	0.7383	0.8189	0.4590	0.5258	0.2421	0.2369	0.1322	0.0334	0.0333	0.0318
System lifetime (sec)	194.5	194.6	194.7	193.1	196.0	196.0	196.0	219.2	213.9	212.6
Received Pkt #	130146	137292	132674	139351	122956	122968	125424	141262	137685	137398

Table 1. Compensated results of optimal flow control and best results of AODV and load balanced routing. "Retrans" denotes the results obtained with the retransmission effect compensated, and "Overhearing" denotes the results obtained with both the retransmission and overhearing effects compensated.

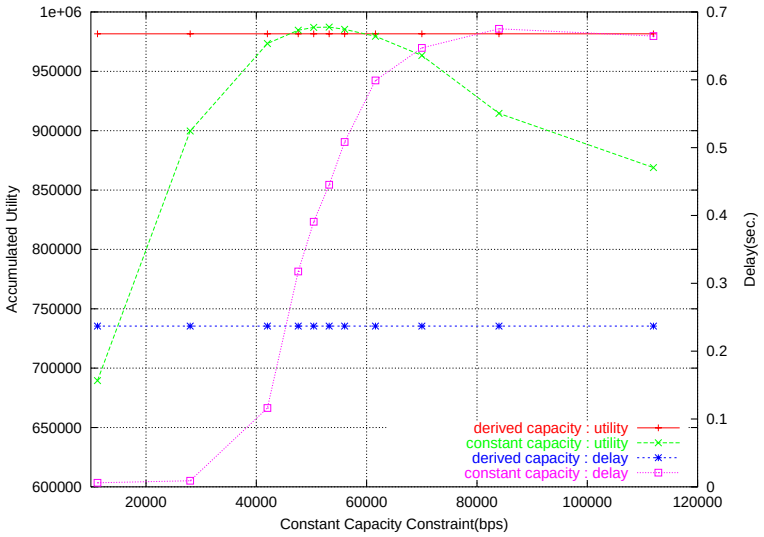


Fig. 9. Performance comparison between the energy aware flow control solution obtained by using the network capacity derived in Section 4 to that obtained by using arbitrarily chosen values.

change the first term in Eq. (3) from $\sum_{k:i\in N_k}\sum_{s\in S_n} q_{ki}^{(s)}$ to $\sum_{k:k\in N_i}\sum_{j:j\in N_k}\sum_{s\in S_n} q_{kj}^{(s)}$. Fig. 8 gives the compensated results along with the best results achieved in AODV and balancing routing. Table 1 gives the corresponding numerical values. With both effects compensated, the utility value remains roughly the same but the delay is much improved and the gap between the simulation result and the theoretical value becomes smaller.

The necessity of deriving the node capacity: To investigate whether or not deriving the network capacity, C_i, for each node i is truly necessary, we compare the energy-aware flow control solution obtained by using the network capacity derived in Section 4 to that obtained by using arbitrary values of C_i. Fig. 9 gives the simulation results. The scenario with the network capacity $C_i = 53$ kbps for all nodes achieves the maximum utility. The result in the accumulated utility deteriorates if either a more or less stringent node capacity constraint is imposed. The solution obtained by using the network capacity derived in Section 4 achieves slightly lower utility but also incurs a smaller end-to-end latency. Without exhaustively testing all the possible values of the network capacity, the conservative network capacity derived in Section 4 provides a reasonable setting to the problem.

The effect of more stringent constraints on the performance: As discussed above, given the above parameter setting for energy consumption, the energy constraint is not so stringent as compared to the node capacity constraint. As a result, the energy-aware flow control solution usually lies at the "boundary" imposed by the node capacity constraint, which in turns causes instability to the system. To further study the effect of these constraints on the performance, we now impose more stringent constraints on the node capacity or the energy consumption. Specifically, the node capacity in Eq. (4) is changed from C_i to αC_i, where α varies from 0 to 1, and the energy pa-

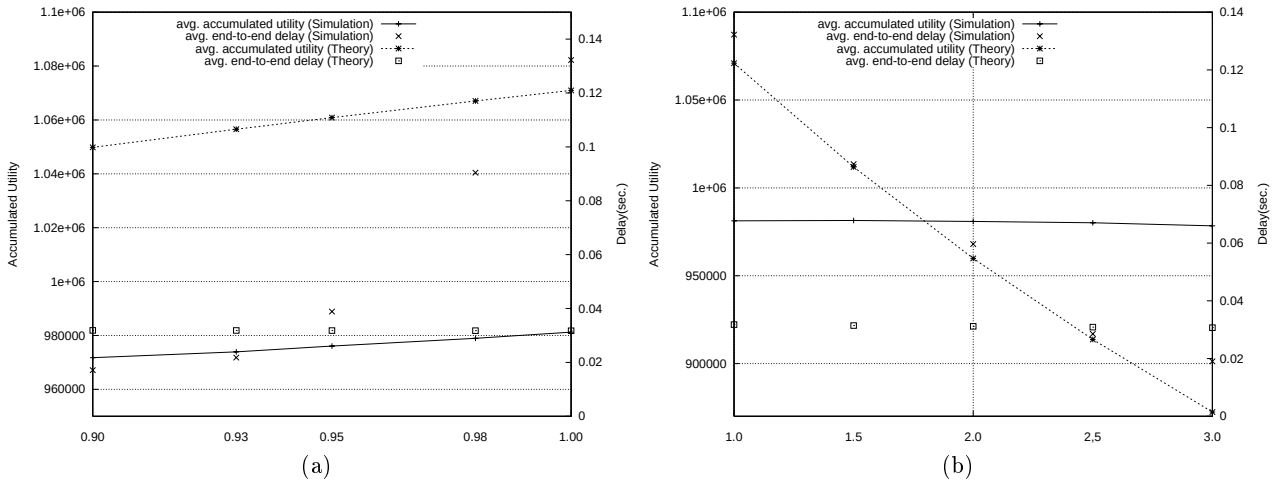


Fig. 10. The accumulated utility and the end-to-end delay obtained from simulation and theoretical formulation with respect to α ((a)) and β ((b)). The smaller the value of α, the more stringent the node capacity constraint. The larger the value of β, the more stringent the energy constraint.

rameters in Eq. (3) are changed from e_r and e_s to $\beta \cdot e_r$ and $\beta \cdot e_s$, where $\beta > 1$. Fig. 10 gives the accumulated utility and end-to-end delay with respect to α (Fig. 10 (a)) and β (Fig. 10 (b)). Despite slight reduction in the accumulated utility is observed for both the simulation and theoretical results, the end-to-end delay incurred in the simulation decreases significantly as the value of α decreases. Also, as shown in Fig. 10(b), although the accumulated utility derived using the theoretical formulation decreases significantly as the value of β increases, the accumulated utility obtained in the simulation remains essentially at same value. Similar to the effect of constraining node capacity, the end-to-end delay decreases dramatically as a more stringent energy constraint is imposed.

As the energy constraint is less stringent than the node capacity constraint, we investigate the effects of removing the energy constraint in the second case introduced in Sec. 3. That is, the objective function does not include the system lifetime and the energy constraint is removed. We again change the node capacity in Eq. (4) from C_i to αC_i, with $\alpha = 0.95$, 0.98 and 1.00. The last column of Fig. 8 gives the result. The accumulated utility is only slightly smaller than that in the solution that considers the energy constraint. This again results from the choice of the utility function and parameters of energy consumption.

6 Conclusion and Future Work

In this paper, we formulate the problem of data transport in sensor networks as an optimization problem whose objective function is to maximize the amount of information (utility) collected at sinks (subscribers), subject to the flow, energy and channel bandwidth constraints. Also, based on a Markov model extended from [3], we derive the link delay and the node capacity in both the single and multi-hop environments, and figure them in the problem formulation. We study three special cases under the problem formulation. In particular, we consider the energy-aware flow control problem, derive an energy aware flow control solution, and investigate via *ns-2* simulation its performance.

The simulation results indicate that the solution to the energy-aware flow control problem achieves highest accumulated utility as compared to AODV and balanced routing with the use of various source rates. However, due to the use of logarithmic utility functions and energy parameters given in [6], the performance improvement that results from using the energy-aware flow control solution is not significant. On the other hand, the optimal solution does achieve both high utility and low end-to-end delay if more stringent node capacity and/or energy constrains are imposed.

We will continue our research along three thrusts. First, we would like to consider both the flow control and routing problem simultaneously and solve the general formulation in Eqs. (1)-(4). One serious drawback of the above optimization problem is that the search space is so broad and the control variables, $q_{ij}^{(s)}$, could be of the order of $O(N^3)$ if no constraints are imposed. Several heuristic rules can facilitate to eliminate unlikely solutions and reduce the search space. For instance, based on geographical information, $q_{ij}^{(s)}$ is considered only when nodes i and j are within the radio transmission range and node j is close to one of sinks than node i for the source s. With use of this rule, the order of control variables is reduced to $O(N^2)$. Second, the applications considered in the simulation study are monitoring applications in which the sending data is static. We would like to apply the utility based approach to more dynamic applications such as the target tracking system. Finally, by approximating certain aspects of the general formulation, we would like to develop a distributed version of the utility based approach.

References

1. C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.
2. D. P. Bertsekas, "Nonlinear Programming, " 2nd edition, Athena Scientific, Belmont, MA, 1999.
3. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," in IEEE Journal on Selected Area in Communications Vol.18, No.3, March 2000.
4. J. Byers and G. Nasser, "Utility-Based Decision-Making in Wireless Sensor Networks (Extended Abstract)," Proceedings of IEEE MobiHOC 2000, Boston, MA, August 2000, pp. 143-4.
5. J.-H. Chang , L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," INFOCOM 2000
6. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in ACM Wireless Networks Journal, Volume 8, Number 5, September, 2002.
7. R. Horst, P.M. Pardalos, and N.V. Thoai, "Introduction to Global Optimization," 2nd ed. Kluwer Academic Publishers, Dordrecht, Boston, London, 2000.
8. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *In Proc of the Sixth Annual IEEE/ACM International Conference on Mobile Computing and Networking,* August 2000.
9. K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance,"in ACM MOBICOM, San Diego, CA, September 2003.
10. F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," Journal of Operational Research Society, vol. 49, no. 3, pp. 237V252, March 1998.
11. R.-F. Liao and A. T. Campbell, "A Utility-Based Approach to Quantitative Adaptation in Wireless Packet Networks," ACM Journal on Wireless Networks (WINET), Vol. 7, No. 5, pp. 541-557, September 2001.
12. S. H. Low and D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," IEEE/ACM Transactions on Networking, 7(6):861-75, December 1999
13. H. Luo, P.Medvedev, J. Cheng and S. Lu, "A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks," IEEE INFOCOM 2001, Anchorage, April 2001.
14. Y. Qiu and P. Marbach, "Bandwidth Allocation in Wireless Ad Hoc Networks: A Price-Based Approach," in IEEE INFOCOM 2003, San Francisco, April 2003.
15. C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Pricing and Power Control in a Multicell Wireless Data Network," IEEE Journal on Selected Areas in Communications, Vol. 19, No. 10, pp. 1883-1892, Oct. 2001.
16. R.J. Vanderbei and D.F. Shanno, "An interior-point algorithm for non-convex nonlinear programming," Computational Optimization and Applications, 13:231–252, 1999.
17. Y. Xu, J. Heidemann, D. Estrin, "Geography-informed Energy Conservation for Ad-hoc Routing," *In Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (ACM MobiCom)*, Rome, Italy, July 16-21, 2001.
18. Y. Xue, B. Li, K. Nahrstedt, "Price-based Resource Allocation in Wireless Ad Hoc Networks," in the Proceedings of the Eleventh International Workshop on Quality of Service (IWQoS 2003)
19. UCB, LBNL, "VINT network simulator," http://www-mash.cs.berkeley.edu/ns/.
20. J. Wang, L. Li, S. H. Low and J. C. Doyle, "Can TCP and shortest path routing maximize utility," Proceedings of IEEE INFOCOM, San Francisco, April 2003